REMARKS

In summary, claims 1, 3, 7, 10, 11, 14-20, and 27-31 are pending. Claims 1, 3, 7, 10, 11, 14-20, and 27-31 are rejected under 35 U.S.C. § 103. Applicant respectably traverses all rejections. No claims are amended. No matter is added.

<u>Claims Rejections - 35 U.S.C. §103</u>

Claims 1, 3, 7, 10, 11, 14-20, and 27-31 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,158,045, issued to You (hereinafter referred to as "You"), in view of U.S. Patent No. 6,470,388, issued to Niemi *et al.* (hereinafter referred to as "Niemi *et al.*") or U.S. Patent No. 5,533,192, issued to Hawley *et al.* (hereinafter referred to as "Hawley *et al.*").

A *prima facie* case of obviousness has not been established. No articulation of motivation or suggestion to combine You, Niemi *et al.*, and Hawley *et al.* has been provided. Without further explanation, the burden for establishing a *prima facie* case of obviousness has not been met, and Applicant has been denied an opportunity to rebut a *prima facie* case of obviousness.

In the instant Office Action, Examiner states that the test for obviousness "is what the combined teachings of the references would have suggested to those of skill in the art." Applicant submits that Examiner is required to provide some suggestion or motivation to modify or combine the references. And, the initial burden is on the Examiner to provide the suggestion or motivation to combine.

In a rejection under 35 U.S.C. §103, "the reference teachings must somehow be modified in order to meet the claims. The modification must be one which would have been obvious to one of ordinary skill in the art at the time the invention was made." MPEP §706.02

> "To establish a prima facie case of obviousness, three basic criteria <u>must</u> be met. First, there must be some <u>suggestion or motivation</u>, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, <u>to modify the reference or to combine reference teachings</u>. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure. In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). See <u>MPEP § 2143</u> - § 2143.03 for decisions pertinent to each of these criteria." MPEP § 2142
>
> The initial <u>burden</u> is on the examiner to provide some suggestion of the desirability of doing what the inventor has done. (emphasis added)

In the instant Office Action, no articulation of motivation or suggestion to combine You, Niemi *et al.*, and Hawley *et al.* has been provided. For example, at page 6, section 4, of the instant Office Action, it is stated that "It would have been obvious to one of ordinary skill in the art at the time the invention was made for each debuggee to include a debugging type attribute, in addition to the processor attributes (see, for example, column 9, line 17-25), so as to designate a particular debugging type. Such an addition would, for example, enable the engine to perform the designated type of debugging without the need for user input." No indication is provided however, as to why one would modify Niemi *et al.*, and where the motivation or suggestion to modify Niemi *et al.* can be found. To state that the modification of a reference, or a combination of references, achieves a specific result, with nothing more, is conclusory and contrary to current law.

In a recent case (*In Re Kahn*, 441 F.3d 9787 (Fed. Cir. 2006)), the Federal Circuit elaborated on the existing "motivation-suggestion-teaching" requirement for combining references. The Court emphasized that motivation or suggestion to combine or modify <u>must be explained</u>. "[S]ome rationale, articulation, or reasoned basis to explain why the conclusion of obviousness is correct" must be provided. *Id.* at 987. "[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." *Id.* at 988.

As another example, on page 9, section 4, of the instant Office Action, it is stated that "[i]t would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement You with a corresponding debugging type abstraction, such as taught by the Niemi, so as to expressly represent the debugging types that the engine supports (see, for example, you, column 6, lines 31-55), as already intended (see, for example, You, column 5, lines 43-55). Furthermore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the debugging type abstraction in the same manner as the processor abstraction of You, and to organize the debugging type abstraction into a corresponding tree below (see, for example, You, FIG. 9)." Again, no articulation as to the motivation or suggestion to combine You and Niemi *et al.* has been provided.

Nor has any indication been provided as to where motivation or suggestion to combine You, Niemi *et al.*, and Hawley *et al.* can be found. As explained in MPEP § 2143.01, "There are three possible sources for a motivation to combine references: the nature of the problem to be solved, the teachings of the prior art, and the knowledge of persons of ordinary skill in the art." Neither of the three possible sources of motivation to combine You, Niemi *et al.*, and Hawley *et al.* has been implicated in the instant Office Action.

Because no motivation or suggestion to combine You, Niemi *et al.*, and Hawley *et al.* has been provided in the instant Office Action, Examiner has not met the burden for establishing a *prima facie* case of obviousness, and Applicant has been denied an opportunity to rebut any suggestion of motivation to combine. Thus, if Examiner wishes to maintain the claim rejections in view of You, Niemi *et al.*, and Hawley *et al.*, Applicant requests to be provided in a non-final Office Action, an articulation of motivation and/or suggestion to combine You, Niemi *et al.*, and Hawley *et al.* Alternatively, Applicant respectfully requests the rejection of claims 1, 3, 7, 10, 11, 14-20, and 27-31, under 35 U.S.C. § 103, be reconsidered and withdrawn.

Additionally, You, Niemi *et al.*, and Hawley *et al.*, whether considered separately or in any combination, neither disclose nor suggest Applicant's claimed invention. You discloses a debugger built on an object-oriented programming framework that is portable to multiple operating systems and hardware platforms. You teaches with respect to Figure 9 an inheritance graph for sample processor classes in an address class hierarchy. However, as acknowledged by the Examiner, You does not teach a debugging type abstraction hierarchy as claimed. In particular, You does not disclose or suggest a debugger with:

> a plurality of debugging type blocks, each debugging type block for supporting at least one of the plurality of debugging type attributes;
> ...
> wherein a particular debugging type block and a particular processor block are selected for debugging a particular debuggee based on the debugging type attribute and processor attribute of the particular debuggee,
> wherein the plurality of debugging type blocks are organized into a debugging type abstraction available to provide debugging type services that vary in implementation for each debugging type,
> wherein the debugging type abstraction comprises programming code, and wherein at least a portion of the programming code for the debugging type abstraction is common as between at least some debugging type blocks and is shared by such debugging type blocks,
> wherein the programming code for the debugging type abstraction is organized into a tree form with generic code at a base node and more specific levels of code branching out at nodes therefrom, the debugging type abstraction nodes including leaf nodes from which no other nodes branch out, each debugging type block being defined to include a plurality of nodes extending from the base node to a particular leaf node,

as required by independent claims 1 and 20. For such teachings, the Examiner turns to Niemi *et al.* However, combining Niemi *et al.* with You does not overcome the deficiencies of You.

Niemi et al. disclose a system that utilizes "debug" objects as illustrated in Figure 4. Debug sub-class 416 is used to define one or more debug objects 418 for use in debugging an application or process. Each debug object has a level that provides a different granularity of debugging information or control. However, Niemi et al. do not teach that the different levels of debug object comprise programming code wherein:

> at least a portion of the programming code for the debugging type abstraction is common as between at least some debugging type blocks and is shared by such debugging type blocks,
> wherein the programming code for the debugging type abstraction is organized into a tree form with generic code at a base node and more specific levels of code branching out at nodes therefrom, the debugging type abstraction nodes including leaf nodes from which no other nodes branch out, each debugging type block being defined to include a plurality of nodes extending from the base node to a particular leaf node,

Applicant respectfully submits that Niemi *et al.*'s teachings of each level of debug object providing a "different granularity of debugging information or control" does not teach or suggest that the different debug objects share programming code organized into a tree and leaf nodes as claimed or that such code is shared by the different debug objects accessible by debugging type attribute as claimed. Applicant can find no such teachings or suggestions by Niemi et al., and the Examiner has not pointed to any.

Accordingly, even if the teachings of Niemi *et al.* could have been combined with the teachings of You as suggested in the instant Office Action, the claimed invention would not result because neither You nor Niemi *et al.* discloses or suggests that a debugging type abstraction, accessible by specifying a debugging type attribute, may be employed and structured in the manner recited in claims 1 and 20, and that a corresponding processor abstraction likewise be employed and structured in the manner recited in claims 1 and 20. Accordingly, the combination of the You and Niemi *et al.* does not result in the subject matter recited in claims 1 and 20. Applicant respectfully requests reconsideration and withdrawal of the rejection of claims 1, 3, 7, 10, 11, 14, 15, 18-20, and 27-29 over the teachings of You and Niemi et al.

The teachings of You are also deficient with respect to the claimed tree form. In the instant Office Action, Figure 9 of You is referenced. However, Figure 9 shows an addressing abstraction utilized to facilitate the use of target memory addresses in a portable fashion (Abstract), and only a particular one of the nodes within such addressing abstraction is selected to locate a particular address, depending on operating system and/or platform. In contrast, the present invention as recited in claims 1 and 20 requires that the programming code for both the debugging type abstraction and the processor abstraction is organized into a tree form with generic code at a base node and more specific levels of code branching out at nodes therefrom such that each respective block is defined to include a plurality of nodes extending from the base node to a particular leaf node. Moreover, the tree of Figure 9 of You has a plurality of nodes, each for a particular type of address, where selecting a particular node is necessary for selecting the corresponding type of address. In contrast, the tree recited in claims 1 and 20 has a plurality of nodes where a particular debugging type block or processor block is defined by selecting a plurality of such nodes extending from a base node to a particular leaf node, as may best be appreciated with reference to Figures 4 and 5 of the present application. Contrary to the assertions in the instant Office Action, such distinctions are not taught by You.

With respect to claims 16, 17, 30, and 31, the Examiner further acknowledges that You does not disclose that the executable code includes an attribute for use in selection of a particular debugging type block in the engine. For such a teaching, the Examiner cites Hawley et al. for its teaching of the selection of one of multiple debuggers. However, as with You and Niemi *et al.*, the teachings of Hawley *et al.* are deficient in that Hawley *et al.* also fails to teach debugging type abstraction, accessible by specifying a debugging type attribute, may be employed and structured in the manner recited in claims 1 and 20, and that a corresponding processor abstraction likewise be employed and structured in the manner recited in claims 1 and 20. Absent such teachings, even if the cited references could have been combined as proposed by in the instant Office Action, the claimed invention would not have resulted.

In view of the foregoing arguments and remarks, Applicant requests that the rejection, under 35 U.S.C. § 103, of claims 1, 3, 7, 10, 11, 14-20, and 27-31 be reconsidered and withdrawn.

CONCLUSION

It is requested that the forgoing arguments, remarks, and amendments be entered. In view of the foregoing arguments, remarks, and amendments, it is respectfully submitted that this application is in condition for allowance. Reconsideration of this application and an early Notice of Allowance are respectfully requested. In the event that the Examiner cannot allow this application for any reason, the Examiner is encouraged to contact the undersigned attorney to discuss resolution of any remaining issues.

Date: May 11, 2007

/Joseph F. Oriti/
Joseph F. Oriti
Registration No. 47,835

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439